



UPDM and SysML for Systems Engineers

Graham Bleakley, IBM

graham.bleakley@uk.ibm.com

Rational Software

© Copyright IBM Corporation 2010

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 2012		2. REPORT TYPE		3. DATES COVERED 00-00-2012 to 00-00-2012	
4. TITLE AND SUBTITLE UPDM and SysML for Systems Engineers				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) IBM Corporation,1 New Orchard Rd,Armonk,NY,10504				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES Presented at the 2012 DoD Enterprise Architecture Miami, FL, April 30 - May 3, 2012, Government or Federal Purpose Rights License					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Same as Report (SAR)	18. NUMBER OF PAGES 23	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

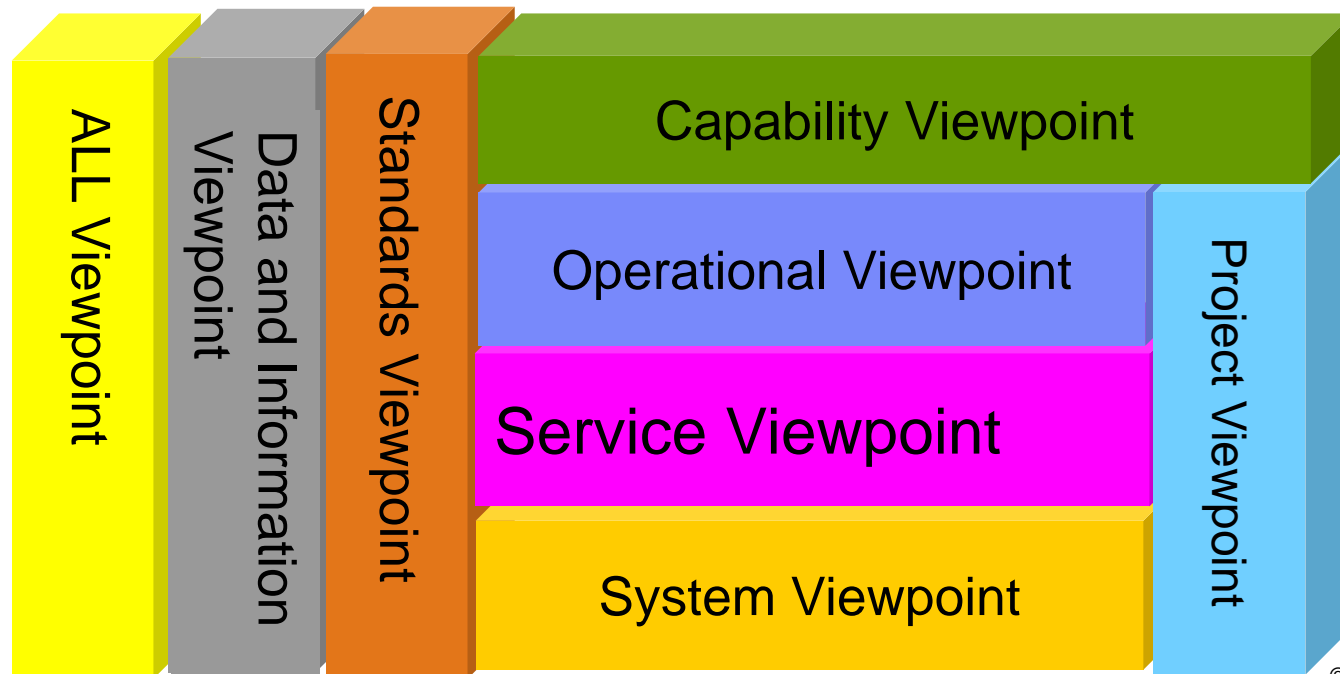
Agenda

This presentation will cover

- Describe how the views are related
- Identify the main viewpoints relevant for the Enterprise Architecture and Systems Engineering phases of development
- Show the patterns across the views and Viewpoints
- Discuss a generic workflow
- Points for discussion on UPDM/SysML modelling

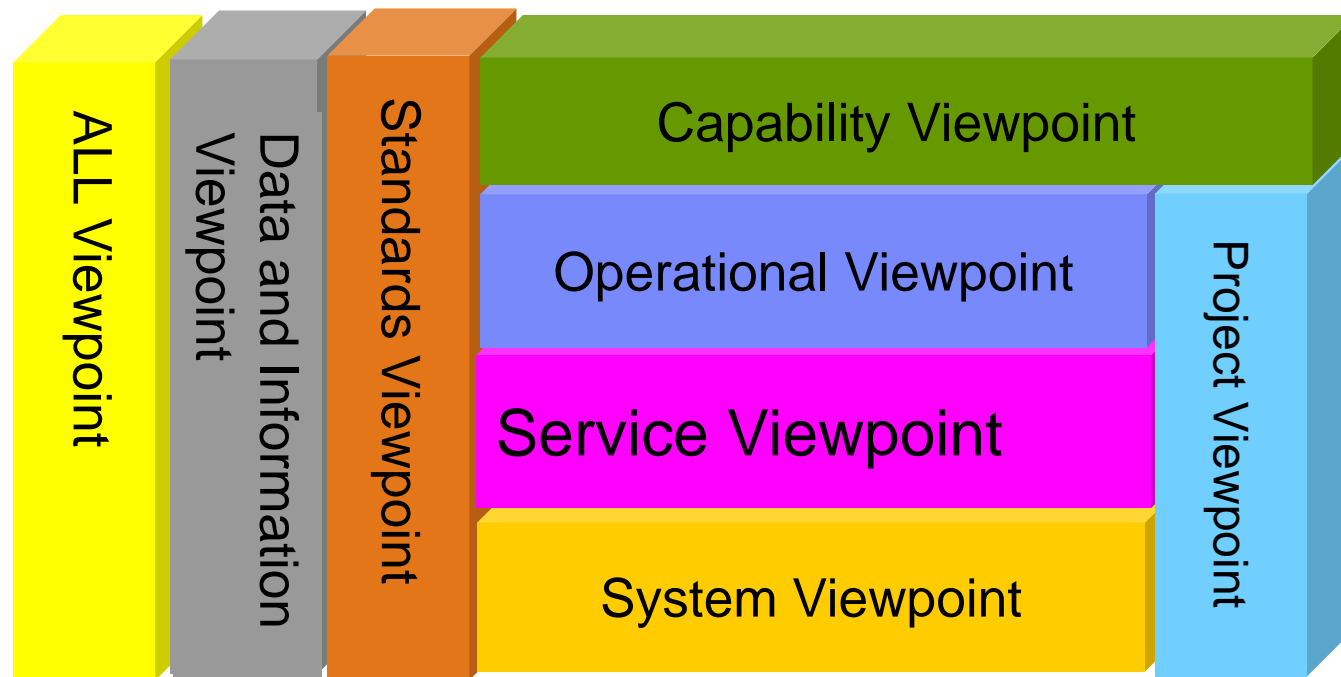
DoDAF 2.0 Viewpoints

- **Capability Viewpoint**:- Capture the capabilities that the enterprise is expected to realise and shows how they are deployed to organizations (prefix CV)
- **Operational Viewpoint**:- Contains the views required to describe the Operational and high level functional aspects of the architecture (prefix OV)
- **Service Viewpoint**:- Captures the views required to specify of the Services (i.e. interface, no implementation) required to support the Operational objectives of the architecture (prefix SvcV)



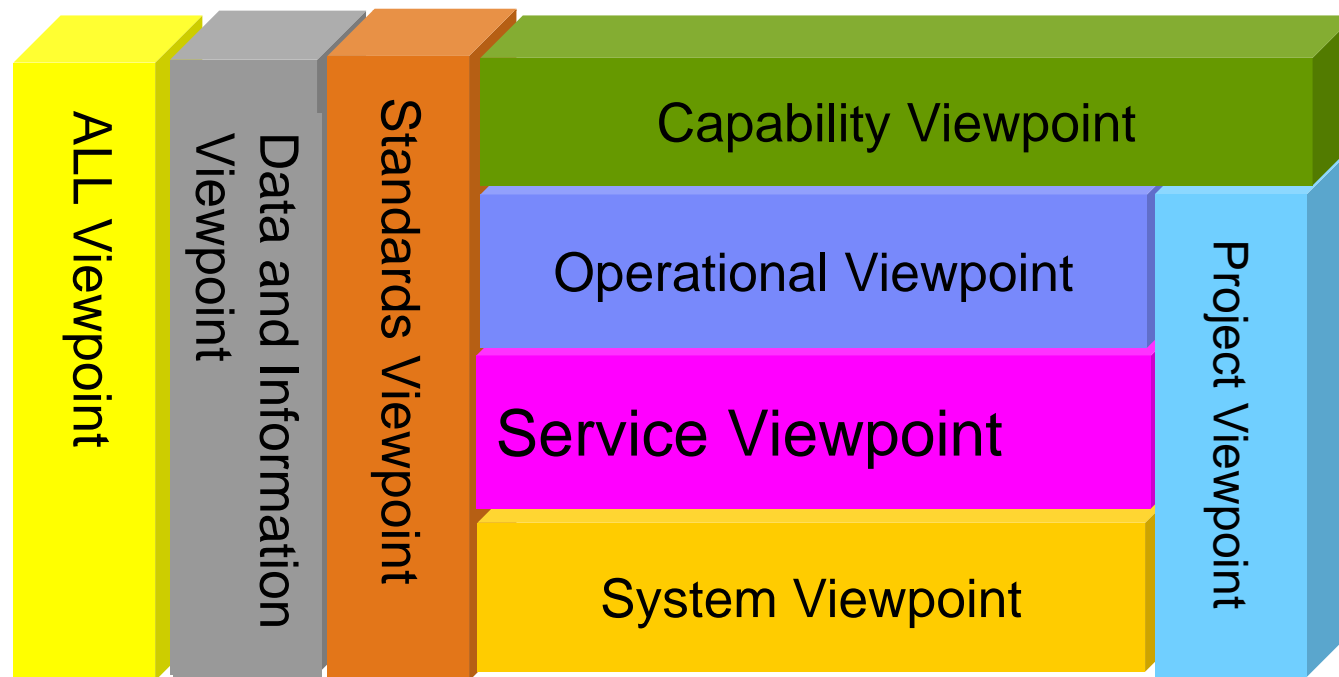
DoDAF 2.0 Viewpoints

- Systems Viewpoint:-Captures the specification of the Systems that are required to be implemented or that already exist that help achieve the operational objectives (prefix SV)
- Project Viewpoint:- Maps the enterprises to the projects and organisations that will realise the capabilities through the development of systems and services (prefix PV)



DoDAF 2.0 Viewpoints

- Standards Viewpoint:-Contains all the information relating to the standards that constrain the other 3 viewpoints (prefix by StdV)
- All Viewpoint:- Contains the summary information about the architecture being developed including the data dictionary (prefix by AV)

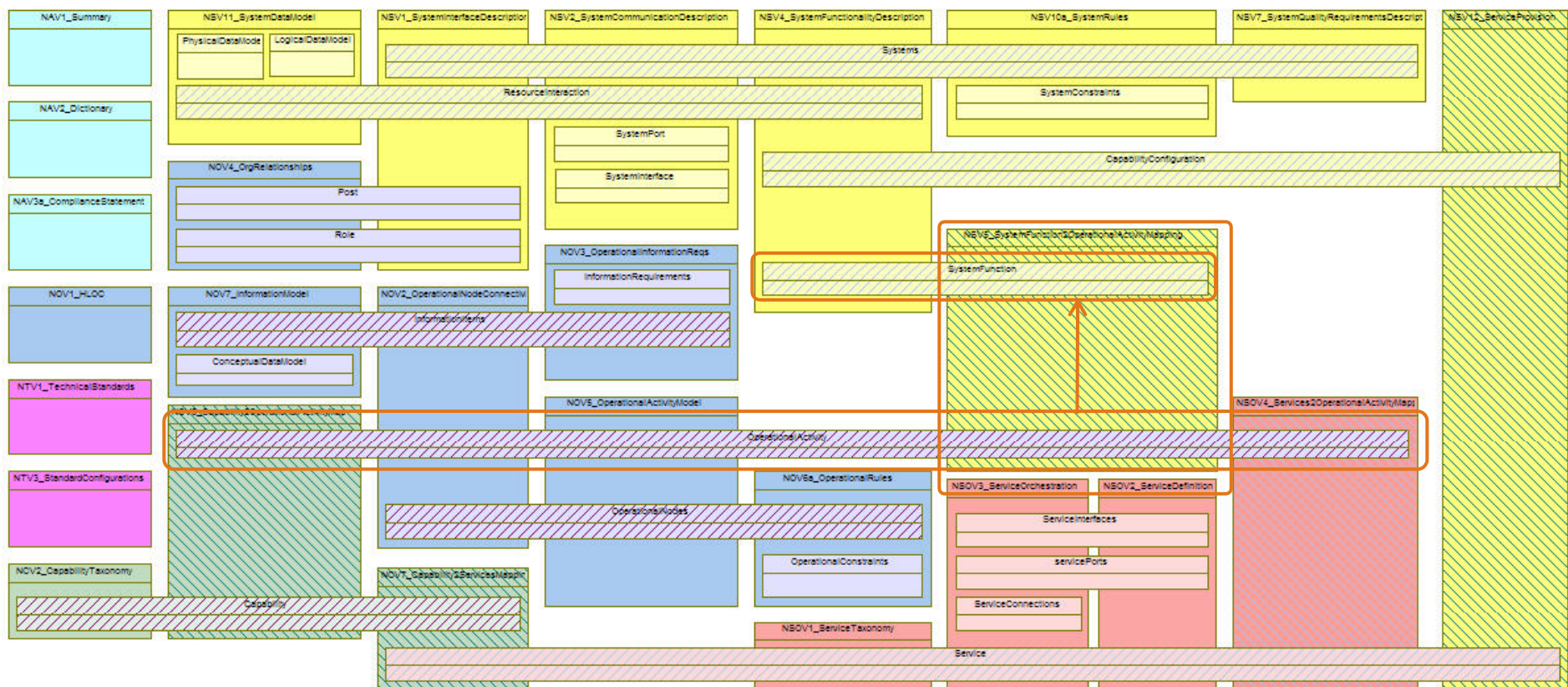


Model Element interaction Across Views

Separation of Elements and Views/Viewpoints

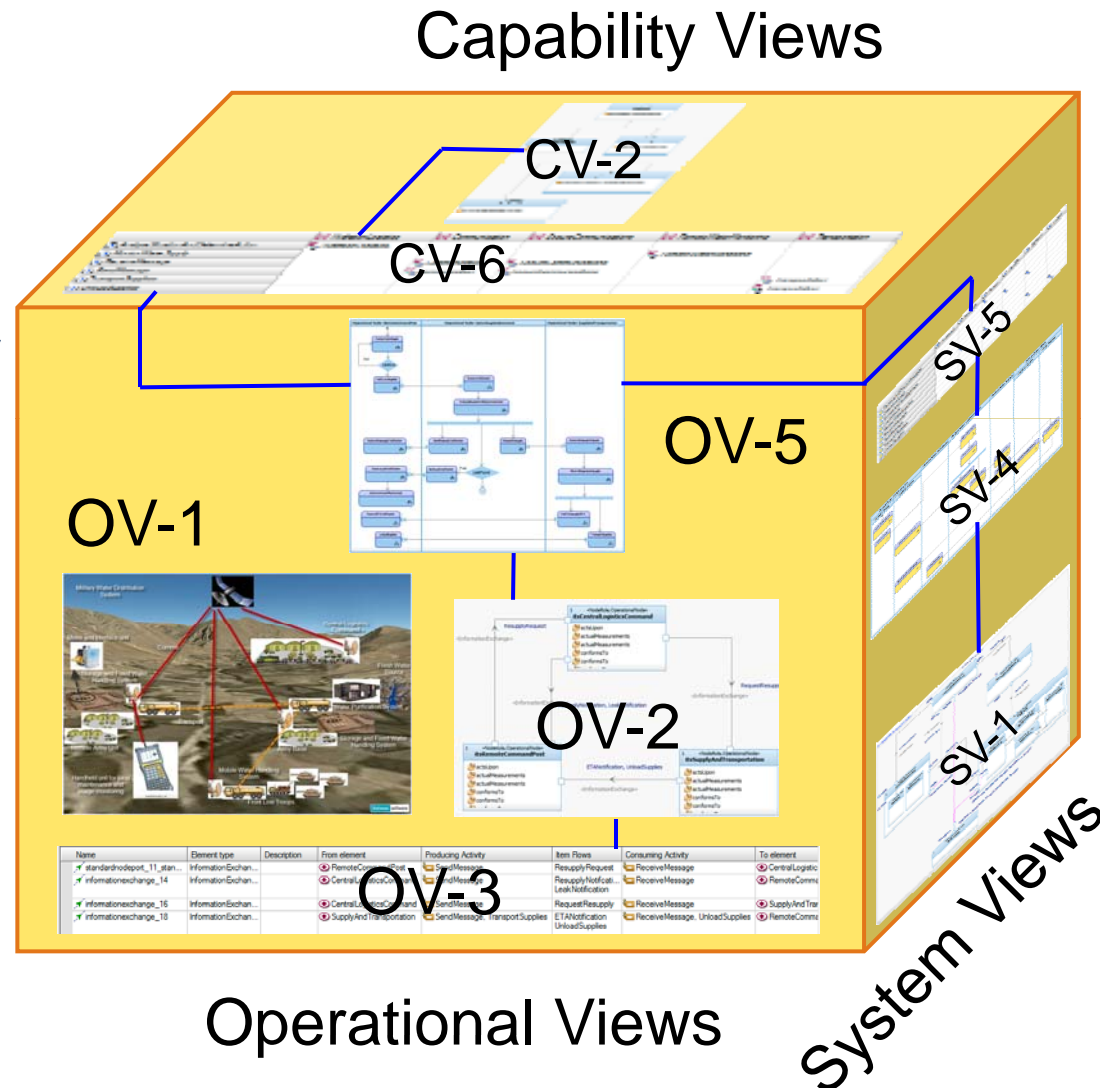
- Some elements seen in multiple views
- Some views ties the viewpoints together e.g. SV-5

Essential Views and model elements based upon Initial Guidelines for use, NAF 3 V1.0



View and Viewpoint interrelationships

- The Viewpoints are not independent but are heavily interrelated
- Each face of Cube represents a viewpoint
- Each window a separate view or product
- Model Elements internal to cube used by multiple views
- Views can act as
 - Filters on the information in the architecture (OV-3, SV-5)
 - Diagrams allowing you to create the information that populates the architecture (SV-1, OV-2)

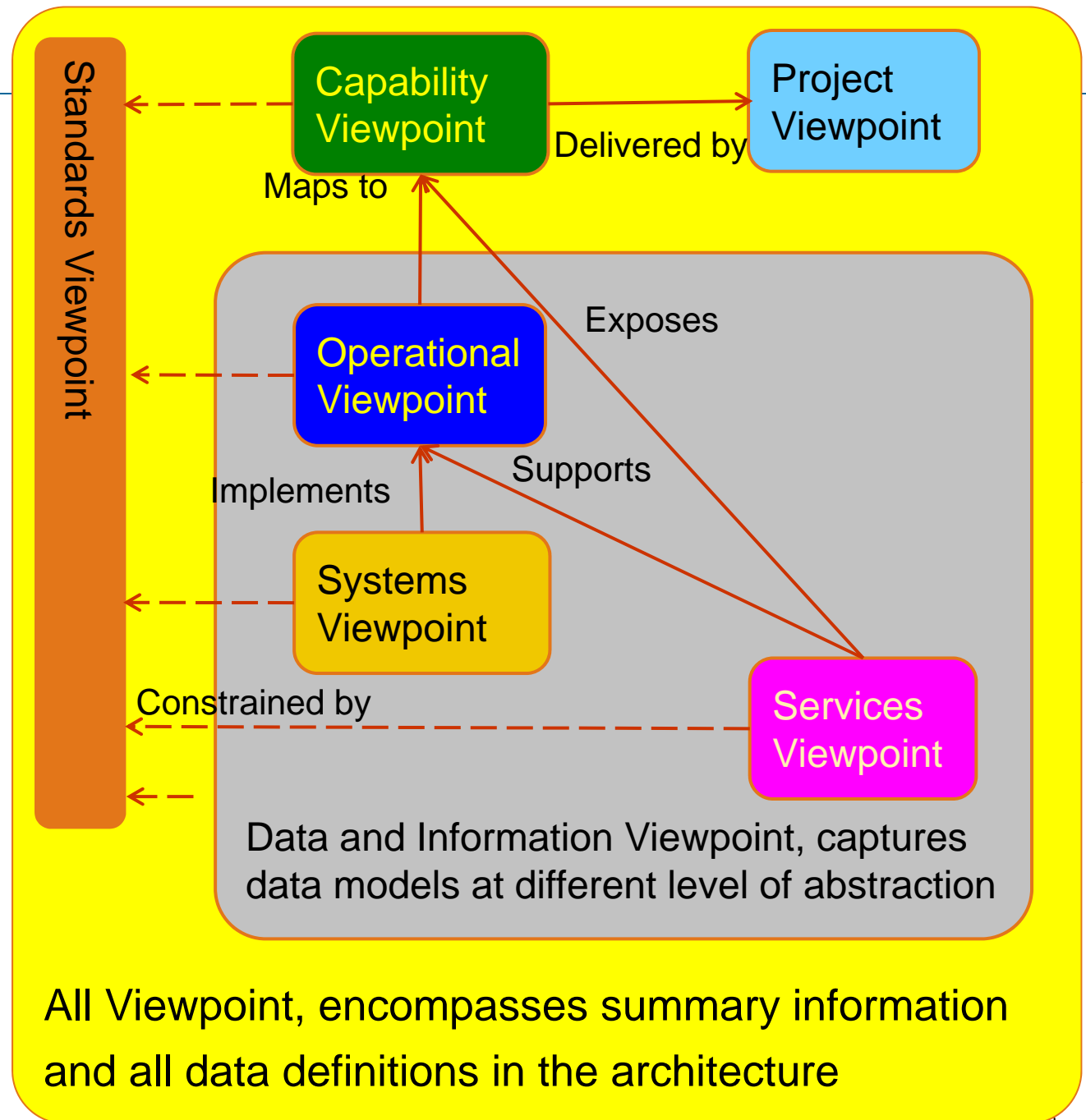


Viewpoint relationships

UPDM DoDAF is a set of traceability matrices

- Systems and Services support and implement Operational elements
- Services expose capabilities (service as an interface)
- Operational elements (activities) map to capabilities
- Capabilities are delivered by Projects
- Systems and Services are realized by projects
- Everything constrained by standards

These relationships are captured in the various matrix views



Enterprise View of UPDM DoDAF

Used by

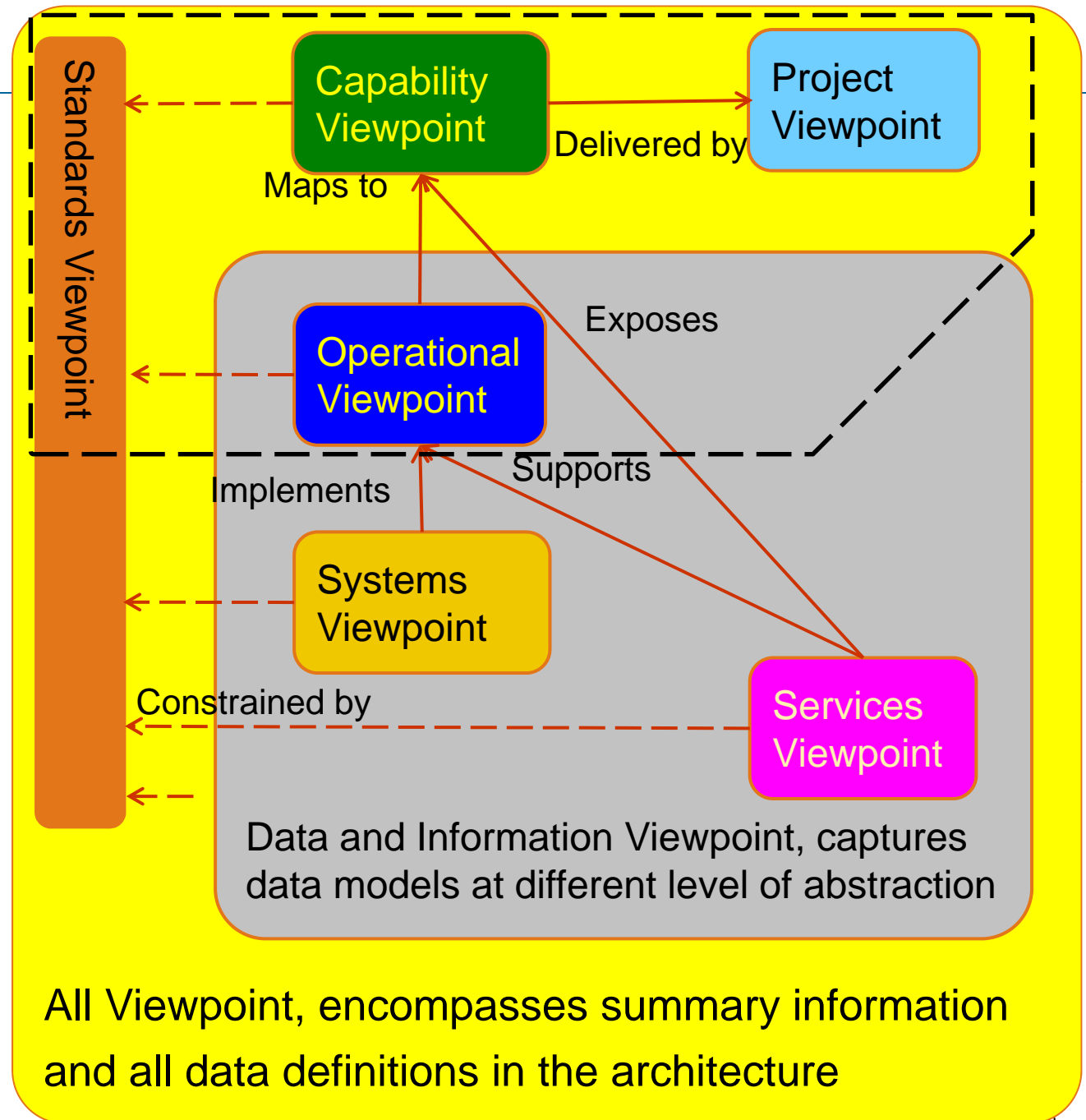
- Planning
- JCIDS
- Operations
- Portfolio management

For

- Capability management
- Operations Planning
- Develop High level requirements for prime suppliers

Still need the

- All view,
- Standards view
- Parts of the Data and Information view



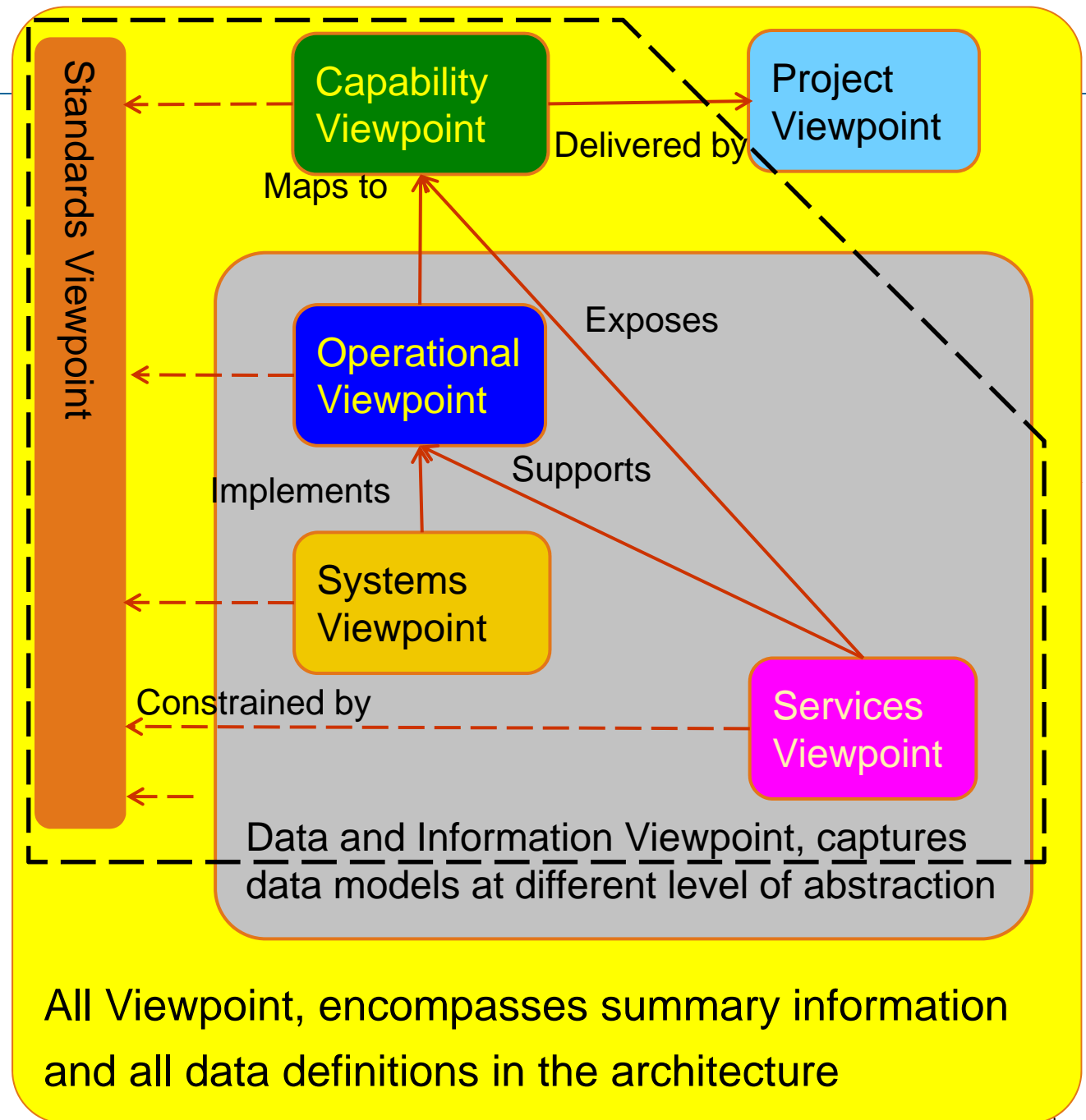
The Systems Engineering View of UPDM DoDAF

Used by

- Portfolio management
- Operations planning
- Defense Acquisition System
- Systems Engineering teams
 - In forces
 - Primes/Tier ones
- Provides requirements to engineering teams

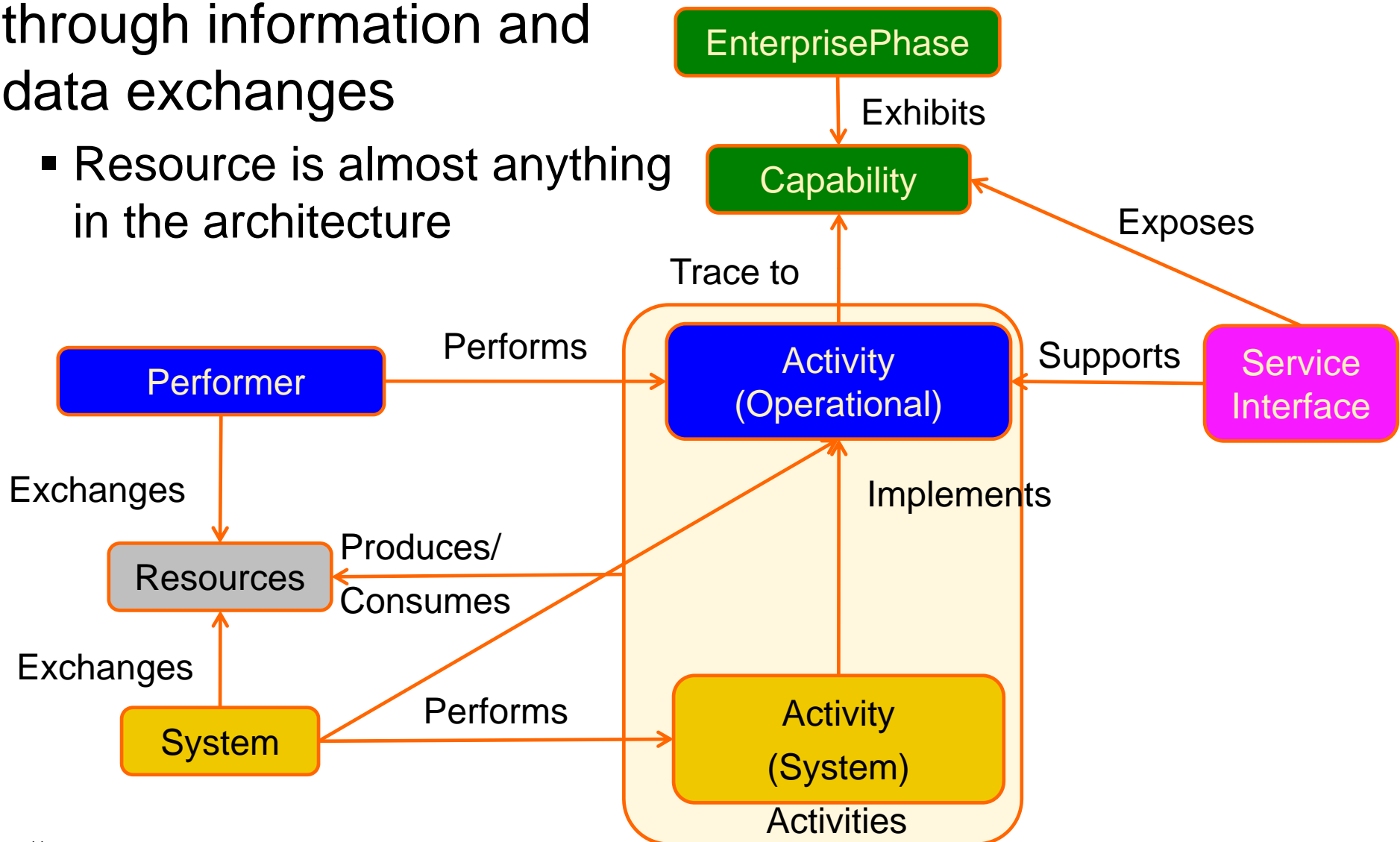
Still need the

- All view,
- Standards view
- Parts of the Data and



Key relationships for Systems Engineering

- Resources exchanged through information and data exchanges
 - Resource is almost anything in the architecture



Patterns in the Framework

Capability

Operational

Systems

Services

Core Views

OV-2

SV-1/2

SvcV-1/2

Structure (IBDs)

OV-5

SV-4

SvcV-4

Behaviour (activity based)

OV-3

SV-3/6

SvcV-3/6

Data exchange Summarys
(Table/Matrix views)

CV-6/7

SV-5a/b

SvcV-5a

Traceability matrices

Supporting Views

OV-6b

SV-10b

SvcV-10b

Behaviour (statemachines)

OV-6c

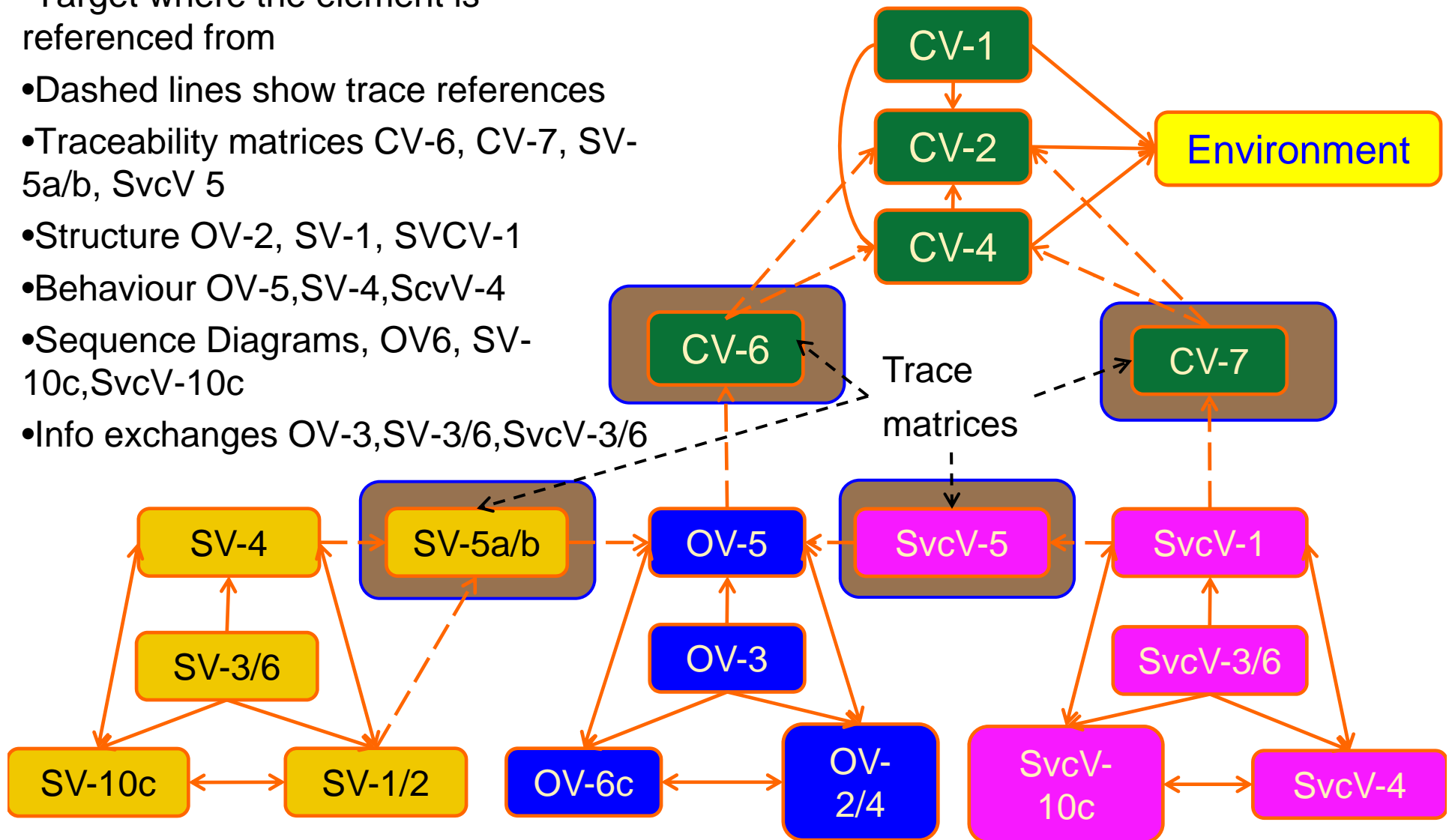
SV-10c

SvcV-10c

Behaviour (sequence diagrams)

Key View dependencies for systems engineering

- Source where element is shown
- Target where the element is referenced from
- Dashed lines show trace references
- Traceability matrices CV-6, CV-7, SV-5a/b, SvcV 5
- Structure OV-2, SV-1, SVCV-1
- Behaviour OV-5,SV-4,ScvV-4
- Sequence Diagrams, OV6, SV-10c,SvcV-10c
- Info exchanges OV-3,SV-3/6,SvcV-3/6



High level workflow (core)

- Develop capability views

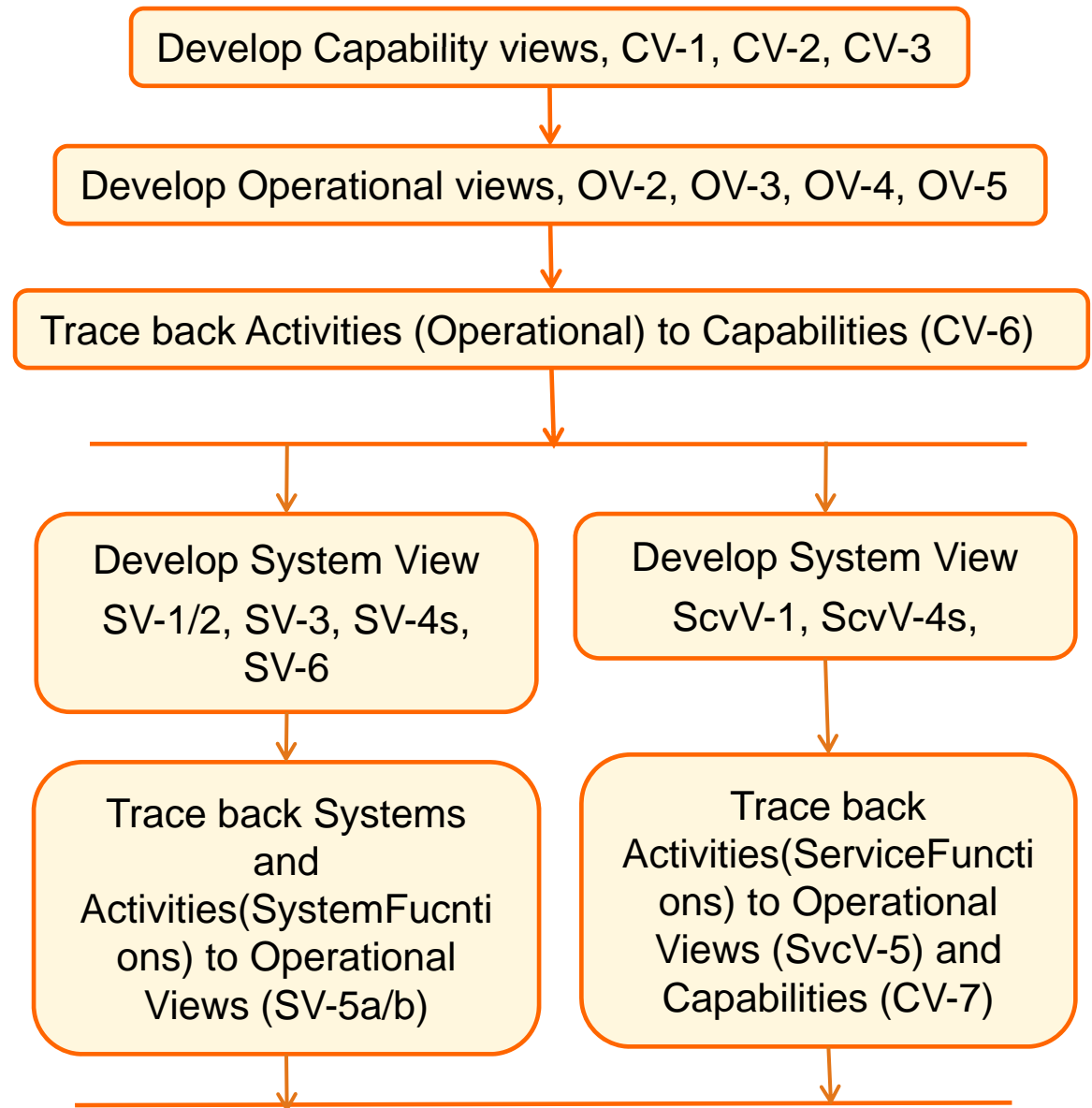
- Enterprise visions
- Capability hierarchy and dependencies

- Develop Operational views

- Behavior, structure,
- Derive data exchange matrices
- Trace back to capabilities

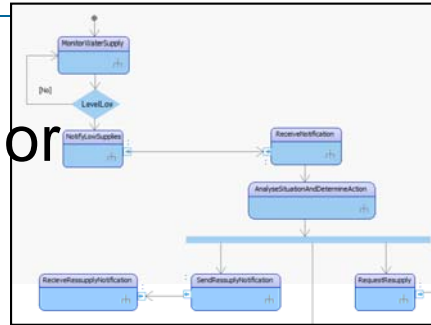
- Develop systems/service views

- Behavior, structure,
- Derive data exchange matrices
- Trace back
 - Systems to operations that they implement
- Trace back Services to the
 - Operations they support
 - Capabilities they expose

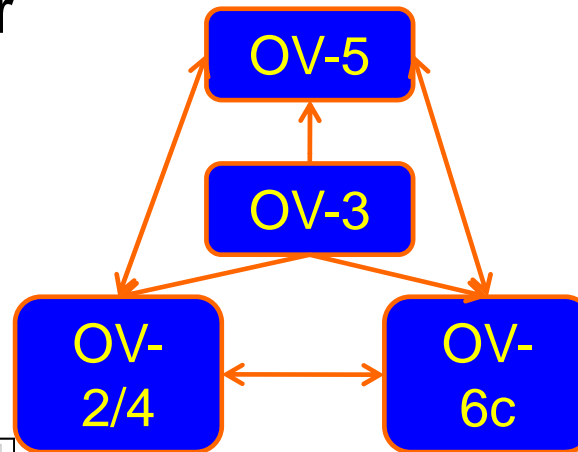
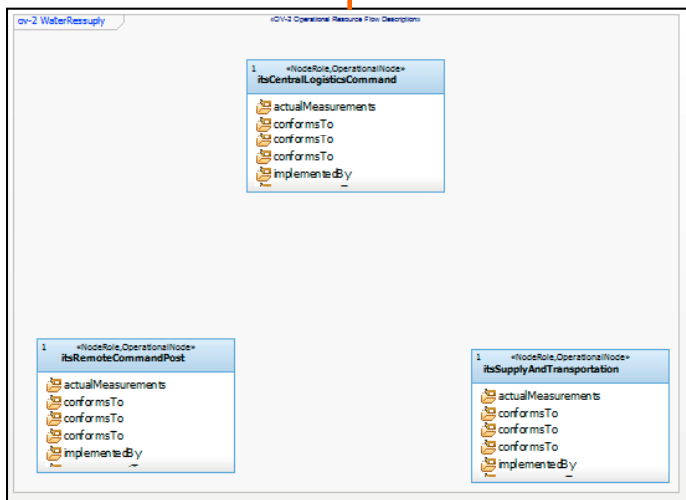


The generic workflow applied to the OV's

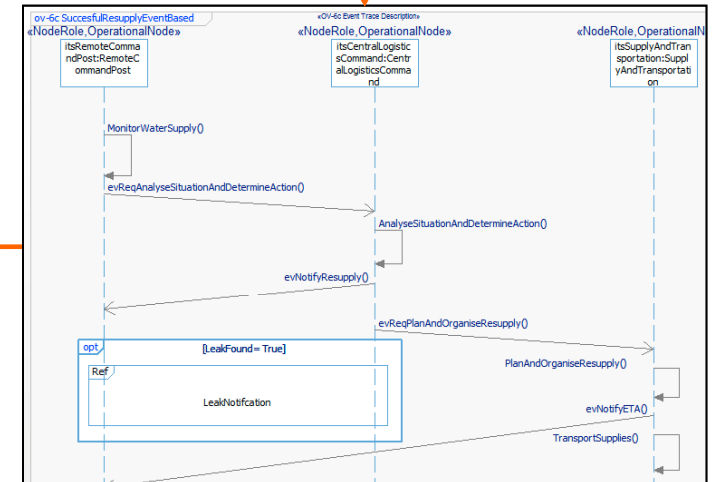
Step-1 BlackBox behavior



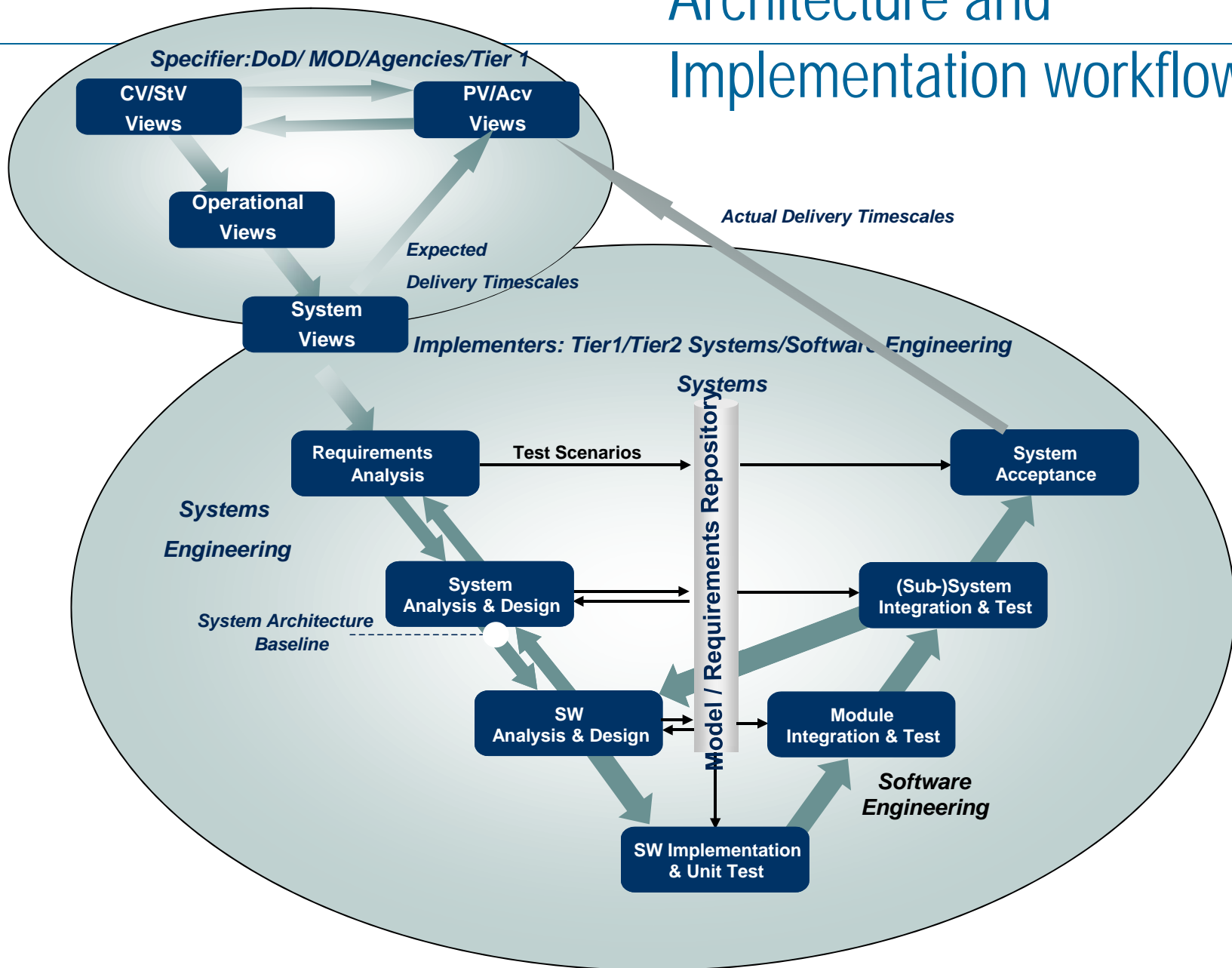
Step-3 Allocate behavior to structure (WhiteBox)



Step-2 Define Structure



Architecture and Implementation workflow

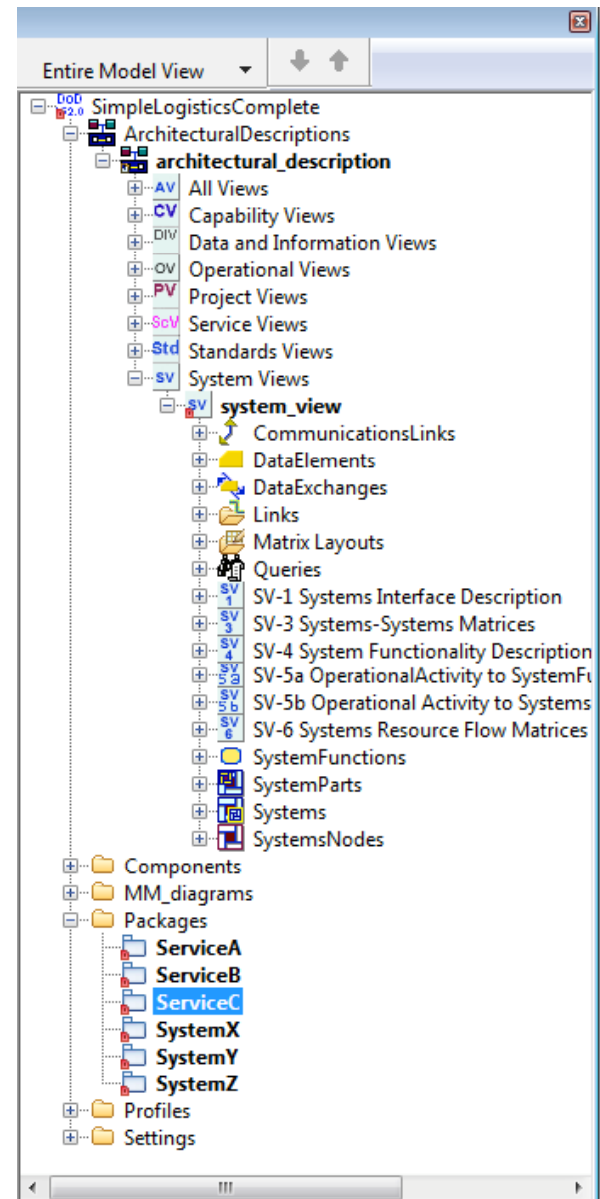


Relating UPDM and SysML models Discussion

- Handling requirements
 - In UPDM requirements driven from the level above
 - Non functional requirements captured as constraints
 - Also work with Parametric diagrams
 - It is possible to import requirements into UPDM models and tie them to Capabilities
- 3 main options
 - Keep SysML models as part of UPDM models
 - Import parts of UPDM models directly into a SysML model
 - Reference UPDM elements from SysML models
- Options as part of these are
 - Embed SysML elements directly in the UPDM elements
 - Ties the elements too closely into UPDM but I think there will be issues
 - Separate out SysML models and trace to UPDM elements
 - Keep to the patterns in UPDM re traceability to the upper levels
- 4 th option
 - Separate UPDM and SysML models and trace through third party tool (i.e. DOORS)

Relating UPDM and SysML models Discussion

- Keep SysML models as part of UPDM models
 - Models become very large and monolithic
 - Hard to maintain and manage

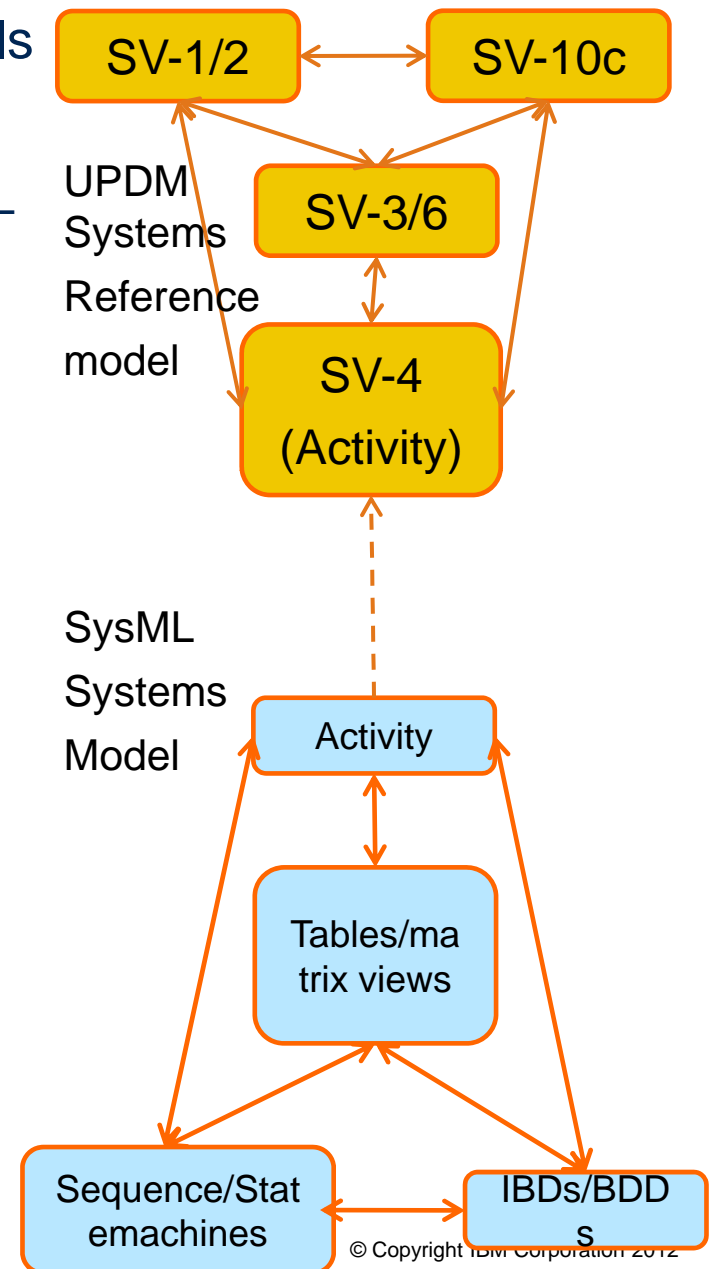


Relating UPDM and SysML models Discussion

- \top Import parts of UPDM models directly into a SysML model
 - Evolve the model with SysML elements
- End up with separate models
- Disconnect between the high level specification of a System in the UPDM model and the \top SysML
- Issues with the need to maintain consistency between the views
- Issues with pushing more detail up into the UPDM views as there will then be inconsistency in the detail and you may lose relations/dependencies to the elements at the higher level
 - **Interchange issue as you are replacing elements you may lose ownership of relationships (might be OK in the same tool, but XMI ?)**
 - **Need Diff-Merge capability for XMI ?**

Relating UPDM and SysML models Discussion

- Reference UPDM elements from SysML models
 - Maintain the consistency of the UPDM model
 - Maintain the patterns of the UPDM structure in the SysML model
 - Trace between SysML and UPDM easily
 - End up with a number of Branch SysML models that reflect the lower level system structure
 - Cannot use the UPDM elements as they are read only (Advantage)
- Initial proposal
 - Combination of taking UPDM models by reference and importing the lower levels of the UPDM architecture and removing UPDM stereotypes
 - Change to SysML elements
 - Trace back to reference elements
 - Gives the possibility of reuse and also shows the reference to original elements



Summary

- Brief review of Viewpoints
- How elements are related across the viewpoints
- Identification of patterns in the view points
- Generic Workflow
- UPDM and SysML